**Q1. Describe the differences between text and binary files in a single paragraph.**

**Q2. What are some scenarios where using text files will be the better option? When would you like to**

**use binary files instead of text files?**

**Q3. What are some of the issues with using binary operations to read and write a Python integer**

**directly to disc?**

**Q4. Describe a benefit of using the with keyword instead of explicitly opening a file.**

**Q5. Does Python have the trailing newline while reading a line of text? Does Python append a**

**newline when you write a line of text?**

**Q6. What file operations enable for random-access operation?**

**Q7. When do you think you&#39;ll use the struct package the most?**

**Q8. When is pickling the best option?**

**Q9. When will it be best to use the shelve package?**

**Q10. What is a special restriction when using the shelve package, as opposed to using other data**

**dictionaries?**

**SOLUTIONS**

***1. Text files and binary files differ in how they store and represent data. Text files store data in ASCII or Unicode characters, where each character is encoded using a fixed number of bytes. On the other hand, binary files store data in a format that is specific to the program or application that created them. Binary files are not human-readable and are often used for storing large amounts of data or complex data structures such as images, audio, or video.***

***2. Text files are suitable for scenarios where the data is in a human-readable format, such as configuration files or log files. Binary files are better for storing complex data structures, such as images or audio, where the data is not in a human-readable format. Binary files are also useful for storing large amounts of data efficiently.***

***3. One issue with using binary operations to read and write a Python integer directly to disk is that the byte order can differ between systems. This can result in data corruption or incorrect data interpretation when the data is transferred between systems with different byte orders. It is also more difficult to read and edit binary data directly compared to text data.***

***4. The with keyword is a context manager in Python that automatically manages the opening and closing of files. Using the with keyword is preferred over explicitly opening and closing a file because it ensures that the file is closed properly, even if an exception occurs. This helps to prevent file leaks and data corruption.***

***5. When reading a line of text in Python, the trailing newline character is included as part of the string. When writing a line of text, Python appends a newline character at the end of the line by default, but this behavior can be overridden using the "end" parameter of the print() function.***

***6. Random-access file operations enable reading and writing data at any location in a file, rather than sequentially from the beginning to the end. The seek() and tell() functions in Python allow random-access file operations by changing the current file position and retrieving the current position, respectively.***

***7. The struct package in Python is used for converting binary data into Python objects and vice versa. It is commonly used for parsing binary file formats, network protocols, and other data structures that use a fixed-size binary format.***

***8. Pickling is a method for serializing Python objects into a binary format that can be stored or transmitted between systems. Pickling is useful for saving the state of a program or for passing data between different Python programs or versions. It is not suitable for storing large amounts of data or for sharing data between programs written in different programming languages.***

***9. The shelve package in Python provides a persistent dictionary that can be stored on disk. It is useful for storing and retrieving large amounts of data that can be indexed by a key, such as dictionaries or lists. The shelve package is designed to work with Python objects, making it easy to store and retrieve complex data structures.***

***10. One special restriction when using the shelve package is that the keys in the dictionary must be strings. This is because the keys are used as filenames for the data files that store the dictionary, and filenames must be strings in most operating systems. Additionally, the data stored in a shelve object must be picklable, meaning that it can be serialized into a binary format.***